**Retrieving data from a number of tables (relations)**

Some tales of a database are logically lined using primary keys and foreign keys.

For example DEPARTMENT tables and EMPLOYEE tables are logically linked using the Primary Key (PK) and Foreign Key (FK).

**DEPARTMENT**

|  |  |  |
| --- | --- | --- |
| **Dept\_No** | **Dept\_Name** | **Block\_No** |
| **1** | **Administration** | **A** |
| **2** | **Accounts** | **E** |
| **3** | **IT HelpDesk** | **A** |
| **4** | **Maintenance** | **C** |
| **5** | **Council** | **E** |
| **6** | **Welfare** | **A** |
| **7** | **Social Science** | **B** |
| **8** | **Computing** | **C** |
| **9** | **Business** | **D** |
| **10** | **Technology** | **E** |

**EMPLOYEE**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Emp\_No** | **Emp\_Fname** | **Emp\_Lname** | **Emp\_Salary** | **Emp\_City** | **Dept\_No** |
| **100** | **Don** | **Dunlop** | **20000** | **Rotorua** | **1** |
| **101** | **Peter** | **Huntly** | 23000 | **Tauranga** | **1** |
| **102** | **Sam** | **Rodrigo** | 40000 | **Rotorua** | **3** |
| **103** | **Richard** | **Bird** | 90000 | **Taupo** | **2** |
| **104** | **Malkeet** | **Singh** | 56000 | **Taupo** | **4** |
| **105** | **Haren** | **Korea** | 78000 | **Rotorua** | **5** |
| **106** | **Ram** | **Malik** | 34000 | **Tauranga** | **6** |
| **107** | **Malan** | **Brando** | 53000 | **Tauranga** | **7** |
| **108** | **Sanka** | **Bradman** | 12000 | **Auckland** | **2** |
| **109** | **param** | **Dunkan** | 120000 | **Auckland** | **2** |

**Example:** We shall list department name and first name of employee using Transact-SQL command. Since these data are coming from two tables, we have to join them temporally within the Transact-SQL.

This line joins two tables temporally.

**The command is:**

use <Your database name>

Select **DEPARTMENT**.Dept\_Name, **EMPLOYEE**.Emp\_Fname

From DEPARTMENT, EMPLOYEE

WHERE **DEPARTMENT**.Dept\_No = **EMPLOYEE**.Dept\_No;

**FK**

**PK**

The temporary table created on the computer memory after joining tables.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **DEPARTMENT .Dept\_No** | **DEPARTMENT .Dept\_Name** | **DEPARTMENT .Block\_No** | **EMPLOYEE .Emp\_No** | **EMPLOYEE .Emp\_Fname** | **EMPLOYEE .Emp\_Lname** | **EMPLOYEE .Emp\_Salary** | **EMPLOYEE Emp\_City** | **EMPLOYEE Dept\_No** |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |

The command works without the words DEPARTMENT & EMPLOYEE

Select **~~DEPARTMENT~~**~~.~~Dept\_Name, **~~EMPLOYEE~~**~~.~~Emp\_Fname

From DEPARTMENT, EMPLOYEE

WHERE DEPARTMENT.Dept\_No = EMPLOYEE.Dept\_No;

Just like this:

Select Dept\_Name, Emp\_Fname

From DEPARTMENT, EMPLOYEE

WHERE DEPARTMENT.Dept\_No = EMPLOYEE.Dept\_No;

But, if you are to list Primary key attribute and Foreign key attribute you need tables names.

Example:

Select DEPARTMENT.Dept\_No, EMPLOYEE.Dept\_No

From DEPARTMENT, EMPLOYEE

WHERE DEPARTMENT.Dept\_No = EMPLOYEE.Dept\_No;

NOTE: JOINING process in a database query is an expensive / time taking process. It involves searching and matching values under the PK and FK. Tables with 500000 records might take about 15 minutes to list the answer to a query. Therefore INDEX files are used to support searching process.

QUESTION: Do we really have to join tables to get results?

Try this command:

Select Dept\_Name, Emp\_Fname

From DEPARTMENT, EMPLOYEE;

**ORDER BY clause**

We use ORDER BY clause to sort a list by an attribute.

Example:

Select Dept\_No, Dept\_Name

From DEPARTMENT

Order by Dept\_Name; -- ascending order

Select Dept\_No, Dept\_Name

From DEPARTMENT

Order by Dept\_Name desc; --Descending order

**GROUP BY Clause**

**WE use** GROUP BY clause on grouping and aggregate calculations.

**Example**

Select Dept\_No, SUM(Emp\_Salary)

From EMPLOYEE

Group by Dept\_No;

SUM is a function used for obtaining totals**.**

**Other Aggregate functions**

**AVG – is used to calculate average**

**COUNT - is used to count records**

**MIN – is used to get the minimum value**

**MAX – is used to get the maximum value**

**Use of LIKE predicate**

**The following command list the departments starting with the letter A.**

select \*

from DEPARTMENT

WHERE Dept\_Name like 'A%'

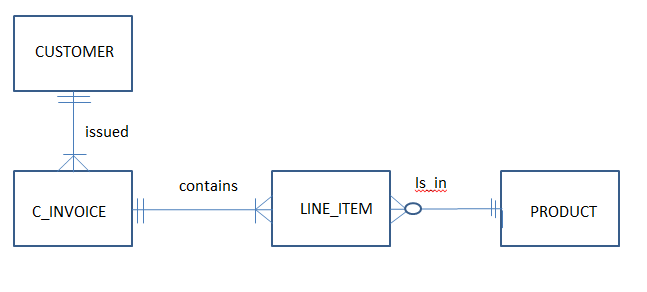
**Please Read**

**http://msdn.microsoft.com/en-us/library/ms173454.aspx**

EXERCISES:

1. List all department names, employee names and their salaries.
2. List the names of employees who work in the “Accounts” department
3. List names of employees who live in Tauranga and their department names.
4. List the employees who work in the Block E.
5. List all department names and Employees names and their salaries sorted by their salaries.
6. List the number employees living in Taupo.
7. List the maximum salary paid for an employee in Rotorua
8. List the department number and the total salary of employees in each department.
9. List the Block number and the total salary of employees in each block.
10. List the Block number and the total salary of employees in each block sorted by block number.
11. List the department names starting with CO.
12. List the departments with more than two employees.

**Write the Transact-SQL code to create the tables in the following ER Model. You must include appropriate date types for the attributes and the constraints for PKs and FKs.**



CUSTOMER(Cno, Cname, Address)

C\_INVOICE (InvNo, Date, Cno)

PRODUCT (Pno, Pname, SalePrice)

LINE\_ITEM(LineNo, InvNo, Pno, quantity)